- 34 -

CLAIMS

We claim:

1.   A computerized method for generating inputs for testing computer
5   programs, the method comprising:

receiving test domain information;

generating a test input according to received test domain information;

invoking a predicate on the test input; and

summing costs assigned to data elements of the test input.

10

2.   The method of claim 1 further comprising determining from an access
list of data elements traversed by the predicate, a last data element accessed in the test
input.

15   3.   The method of claim 2 further comprising varying the last data element
accessed in the test input in response to determining the test input is acceptable.

4.   The method of claim 2 further comprising varying the last data element
accessed in the test input in response to determining the test input is unacceptable.

20

5.   The method of claim 2 wherein the summed costs are costs of paths
through data elements of the test input.

6.   The method of claim 1 wherein the summed costs are costs of all data
25   elements in the input.

7.   The method of claim 1 wherein the summed costs are costs of data
elements accessed by the predicate as it traverses the test input.

8.   The method of claim 1 wherein the summed costs are costs of paths of data elements accessed by the predicate as it traverses the test input.

5   9.   The method of claim 1 wherein the test input is generated as a tree data structure.

10.   The method of claim 9 wherein the summed costs are costs assigned to edges between nodes representing data elements in the tree data structure.

10

11.   The method of claim 1 wherein received test domain information is received via user interaction with a graphical user interface.

12.   The method of claim 11 wherein test domain information received

15   includes an abstract syntax, a predicate, a maximum number of allowable instances for a data type, and a cost indication for an edge between two data types.

13.   The method of claim 1 wherein the predicate fails a tree that has an acceptable maximum allowable summed cost.

20

14.   The method of claim 1 wherein an exceeded summed cost fails a tree that is acceptable to the predicate.

15.   The method of claim 3 wherein the last data element varied, is varied

25   based on a next enumerated data type.

16.   The method of claim 1 wherein the predicate is instrumented with binary code to record an access vector and to sum the costs assigned to data elements.

17.   The method of claim 7 wherein a data element is assigned a zero cost.

18.   The method of claim 7 wherein a cost assignment function assigns
progressively higher costs.

19.   A computer system comprising:
a computer memory having stored therein,
        a module for obtaining test configuration information,
        a module for generating test input based on obtained test configuration
information,
        a module for traversing a generated test input with a predicate,
        a module for summing costs assigned to data elements of the generated
test input, and
        a central processing unit for executing instructions of the modules.

20.   The system of claim 19 wherein the computer memory further has stored
therein a module for saving test cases acceptable to the predicate and summed costs.

21.   The system of claim 19 wherein the computer memory further has stored
therein:
        a module for recording data element accesses made in the generated test input;
and
        a module for enumerating to a next data type from a last data type wherein the
last data type was a data type of a data element in the generated test input.

22.   The system of claim 21 wherein the next data type is a last data in a next
test input.

23.   The system of claim 19 wherein the module for obtaining test configuration information obtains assignments of costs to data types, a predicate, and a path maximum cost.

5

24.   A computer readable medium comprising executable instructions for performing a method comprising:

generating a tree of data elements;

invoking a predicate on the tree;

10   recording data elements accessed as the invoked predicate traverses the tree; and

summing costs of data elements accesses.

25.   The system of claim 24 further comprising saving the tree as a string representation if the tree is accepted by the predicate and summed costs do not exceed a

15   maximum cost.

26.   The system of claim 25 wherein the maximum cost is not exceeded by any path through the tree.

20   27.   The system of claim 24 wherein summed costs of data element accesses exceeds an allowable maximum tree cost sum.

28.   The system of claim 19 wherein obtained test configuration information comprises a root data type with plural fields including a left field and a right field, and a

25   field access cost of a same data type is varied based on whether the same data type is accessed by the left or right field.